

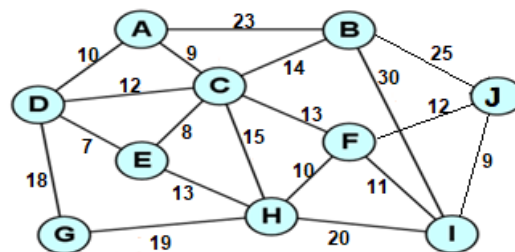
Tugas Kelompok ke-2

Week 4

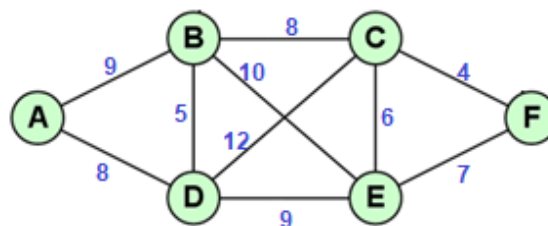
- Berikut ini merupakan permasalahan Knapsack tentukan kombinasi barang yang dapat diambil yang mampu memberikan keuntungan maksimum dengan menggunakan konsep metode Greedy apabila diketahui barang tidak terbatas dan tidak dapat dipecah dimana kapasitas maksimum dari tas adalah 31 Kg.

Barang	Nilai	Bobot
Copper	139587	4
Iron	480331	6
Titanium	698664	9
Bronze	116444	5
Platinum	447144	8
Silver	975218	12

- Buatlah *Minimum Spanning Tree* menggunakan algoritma **Prim-Dijkstra** dan **Kruskal** dari graph berikut!



- Carilah shortest path dari A ke F menggunakan algoritma *Greedy*



4. Dengan menggunakan pendekatan metode ***Greedy***, carilah daftar pekerjaan yang dapat dilakukan untuk memberikan nilai keuntungan paling maksimum!

Pekerjaan	Tenggat Waktu	Keuntungan
A	2	7200
B	3	6150
C	4	3550
D	5	1950
E	1	2850
F	3	2700
G	2	3350
H	4	5600
I	6	6250
J	4	3200

Anggota Kelompok (Group-6) :

- Ahmad Jordan (2702431532)
- Satria Buana Paksi (2702432661)
- Josafat Raka Verdianto (2702427964)
- Faizal Rifqi (2702440184)
- Nia Anisa Fadhila (2602266952)

JAWABAN:

1. Knapsack Problem jika diketahui barang tidak terbatas dan tidak dapat dipecah dimana kapasitas maksimum dari tas adalah 31 Kg. Maka menggunakan metode

Unbounded Knapsack Problem

Barang	Nilai	Bobot (kg)
Copper	139587	4
Iron	480331	6
Titanium	698664	9
Bronze	116444	5
Platinum	447144	8
Silver	975218	12

Untuk menyelesaikan masalah Knapsack dengan metode Greedy, dapat menggunakan langkah-langkah berikut:

- I. Hitung nilai per satuan berat (nilai per kilogram) untuk setiap barang.
 - Copper: $139587 / 4 = 34896.75$
 - Iron: $480331 / 6 = 80055.17$
 - Titanium: $698664 / 9 = 77629.33$
 - Bronze: $116444 / 5 = 23288.8$
 - Platinum: $447144 / 8 = 55892.5$
 - Silver: $975218 / 12 = 81268.17$
- II. Urutkan barang-barang tersebut berdasarkan nilai per kilogram secara menurun
 - Silver : 81268.17
 - Iron : 80055.17
 - Titanium: 77629.33
 - Platinum : 55892.5

- Copper : 34896.75
- Bronze : 23288.8

III. Mulai dari barang dengan nilai per kilogram tertinggi, tambahkan barang ke dalam tas sampai kapasitas tas terpenuhi atau tidak ada barang lagi yang dapat dimasukkan.

- Silver: $31 \text{ kg} / 12 \text{ kg} = 2.58$ (ambil 2 unit)
- Iron: $(31 \text{ kg} - 2 * 12 \text{ kg}) / 6 \text{ kg} = 1.17$ (ambil 1 unit)
- Titanium: $(31 \text{ kg} - 2 * 12 \text{ kg} - 1 * 6 \text{ kg}) / 9 \text{ kg} = 1.44$ (ambil 1 unit)
- Platinum: $(31 \text{ kg} - 2 * 12 \text{ kg} - 1 * 6 \text{ kg} - 1 * 9 \text{ kg}) / 8 \text{ kg} = 0.375$ (ambil 0 unit)
- Copper: $(31 \text{ kg} - 2 * 12 \text{ kg} - 1 * 6 \text{ kg} - 1 * 9 \text{ kg} - 0 * 8 \text{ kg}) / 4 \text{ kg} = 0.75$ (ambil 0 unit)
- Bronze: $(31 \text{ kg} - 2 * 12 \text{ kg} - 1 * 6 \text{ kg} - 1 * 9 \text{ kg} - 0 * 8 \text{ kg} - 0 * 4 \text{ kg}) / 5 \text{ kg} = 0.2$ (ambil 0 unit)

Dengan demikian, kombinasi barang yang dapat diambil adalah:

- 2 unit Silver
- 1 unit Iron
- 1 unit Titanium

Keuntungan maksimum yang dapat diperoleh adalah:

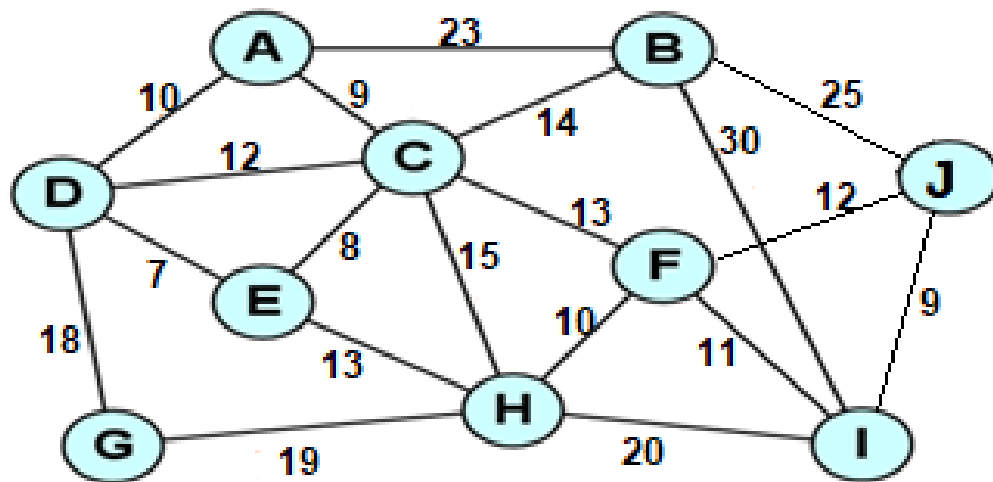
$$2 (975218) + 1 (480331) + 1 (698664) = \mathbf{2927652 \text{ (NILAI)}}$$

2. Berikut minimum spanning tree dengan menggunakan algoritma prim dan algoritma Kruskal

a. Algoritma Kruskal

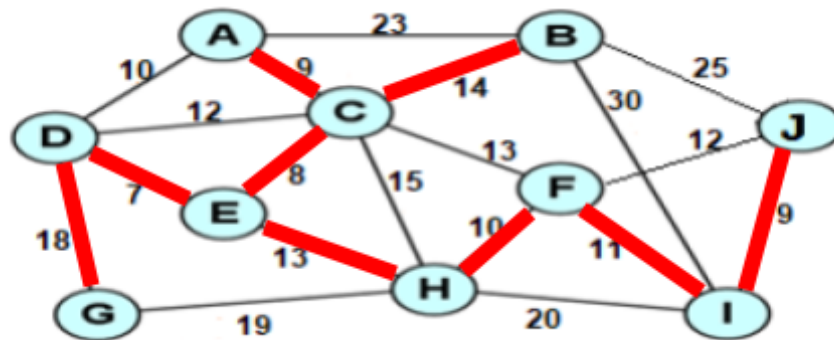
Langkah-langkah :

Mengurutkan sisi dan bobot dari yang terkecil, menggambar kembali graph dengan memperhatikan urutan sisi dan bobot yang telah disusun, apabila saat digambar membentuk looping maka diabaikan.

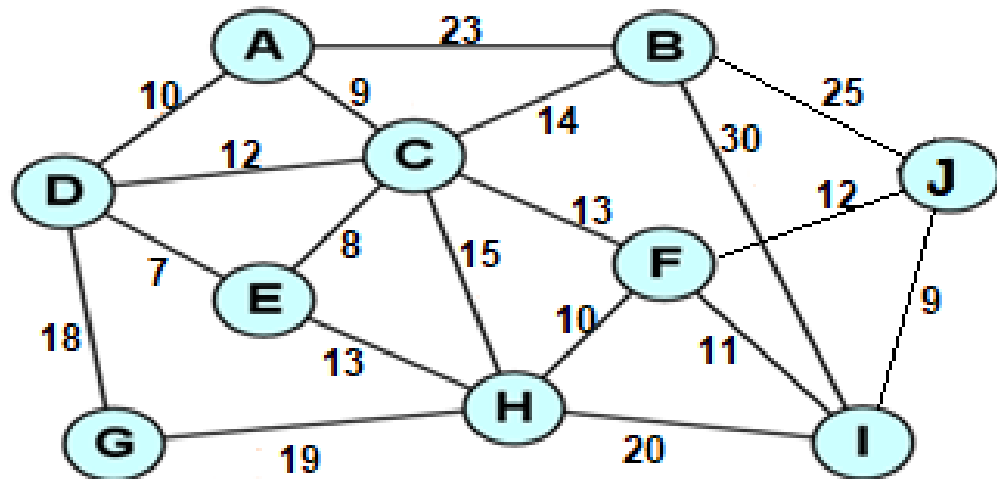


SISI	BOBOT
D,E	7
E,C	8
A,C	9
J,I	9
D,A	10
H,F	10
F,I	11
D,C	12
F,J	12
E,H	13
C,F	13
C,B	14
C,H	15
D,G	18
G,H	19
H,I	20
A,B	23
B,J	25
B,I	30

Berikut minimum spanning tree yang ditandai dengan garis merah :



- b. **Algoritma Prim-Dijkstra**
Langkah-langkah :

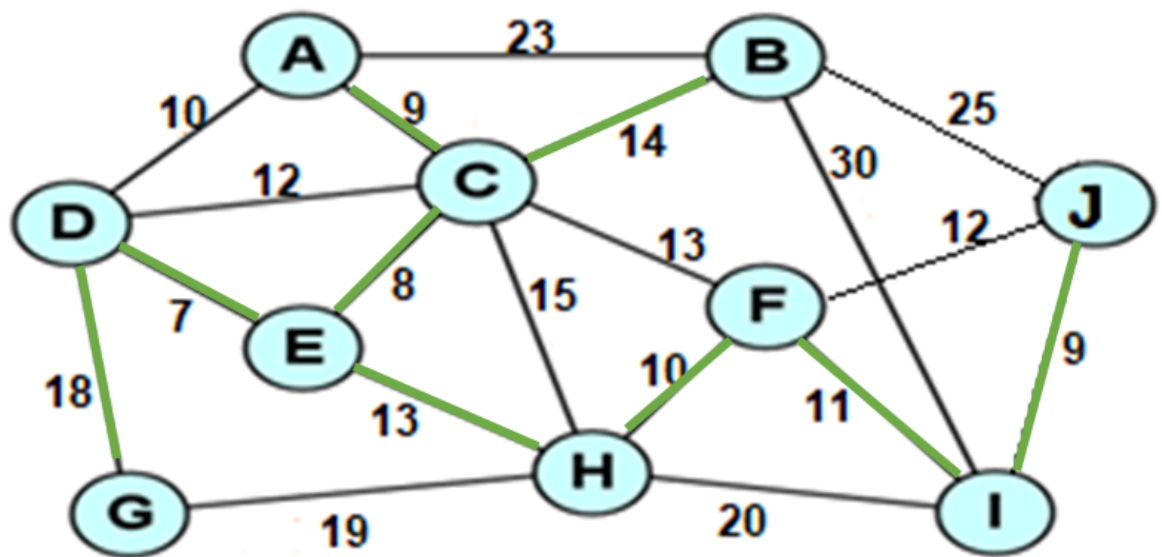


V	A	B	C	D	E	F	G	H	I	J
a	0a	23a	9a	10a	∞	∞	∞	∞	∞	
c		23c	9a	21c	17c	22c	∞	24c	∞	∞
e		23c		21c	17c	22c	∞	24c	∞	∞
d		23c		21c		22c	39d	24c	∞	∞
f		23c				22c	39d	32f	33f	34f
b		23c					39d	32f	53b	48b
h							51h	32f	52h	48b
j							51h		57j	48b
g							51h		57j	
i									57j	

Ditemukan rute menggunakan algoritma Dijkstra dari a - j :
A-C-B-J

c. Algoritma Prim

Dari graph yang disajikan dapat dipilih salah satu titik, kemudian dari titik disebut dapat diperhatikan sisi-sisi yang saling bertemu di titik yang sebelumnya sudah dipilih. Dari sisi-sisi tersebut pilih sisi dengan bobot terkecil, kemudian selanjutnya bertemu di titik dari sisi yang dipilih dan lakukan berulang seperti Langkah sebelumnya. Hindari sisi yang mengakibatkan graph menjadi looping atau berulang. Berikut untuk minimum spanning tree yang ditandai garis hijau:



Langkah	E bersisian	E (W min)	T {E min}
0	-	[7;(D,E)]	[7;(D,E)]
1	[18;(D,G)] [10;(D,A)] [12;(D,C)] [8;(E,C)] [13;(E,H)]	[8;(E,C)]	[7;(D,E)] [8;(E,C)]

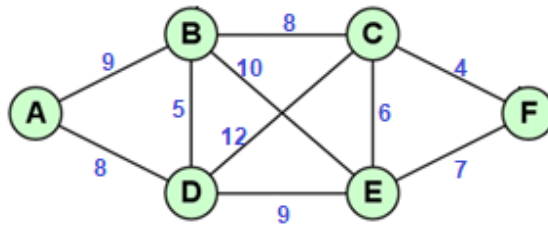
2	[18;(D,G)] [10;(D,A)] [12;(D,C)] [13;(E,H)] [15;(C,H)] [13;(C,F)] [14;(C,B)] [13;(C,F)] [9;(C,A)] [23;(A,B)]	[9;(C,A)]	[7;(D,E)] [8;(E,C)] [9;(C,A)]
3	[18;(D,G)] [10;(D,A)] [12;(D,C)] [13;(E,H)] [15;(C,H)] [13;(C,F)] [14;(C,B)] [23;(A,B)]	[13;(E,H)]	[7;(D,E)] [8;(E,C)] [9;(C,A)] [13;(E,H)]

4	[18;(D,G)] [10;(D,A)] [12;(D,C)] [15;(C,H)] [13;(C,F)] [14;(C,B)] [23;(A,B)] [19;(H,G)] [23;(A,B)] [10;(H,F)] [20;(H,I)]	[10;(H,F)]	[7;(D,E)] [8;(E,C)] [9;(C,A)] [13;(E,H)] [10;(H,F)]
5	[18;(D,G)] [10;(D,A)] [12;(D,C)] [15;(C,H)] [13;(C,F)] [14;(C,B)] [23;(A,B)] [19;(H,G)] [23;(A,B)] [20;(H,I)] [12;(F,J)]	[11;(F,I)]	[7;(D,E)] [8;(E,C)] [9;(C,A)] [13;(E,H)] [10;(H,F)] [11;(F,I)]

	[11;(F,I)]		
6	[18;(D,G)] [10;(D,A)] [12;(D,C)] [15;(C,H)] [13;(C,F)] [14;(C,B)] [23;(A,B)] [19;(H,G)] [23;(A,B)] [20;(H,I)] [12;(F,J)] [9;(I,J)]	[9;(I,J)]	[7;(D,E)] [8;(E,C)] [9;(C,A)] [13;(E,H)] [10;(H,F)] [11;(F,I)] [9;(I,J)]
7	[18;(D,G)] [10;(D,A)] [12;(D,C)] [15;(C,H)] [13;(C,F)] [14;(C,B)] [23;(A,B)] [19;(H,G)] [23;(A,B)] [20;(H,I)]	[14;(C,B)]	[7;(D,E)] [8;(E,C)] [9;(C,A)] [13;(E,H)] [10;(H,F)] [11;(F,I)] [9;(I,J)] [14;(C,B)]

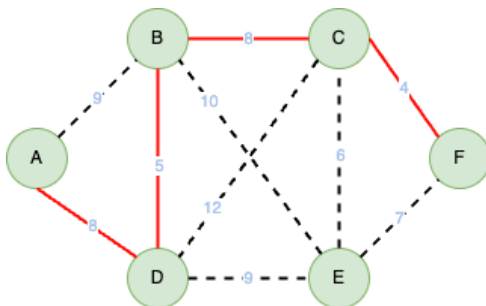
	[12;(F,J)]		
8	[18;(D,G)]	[18;(D,G)]	[7;(D,E)]
	[10;(D,A)]		[8;(E,C)]
	[12;(D,C)]		[9;(C,A)]
	[15;(C,H)]		[13;(E,H)]
	[13;(C,F)]		[10;(H,F)]
	[23;(A,B)]		[11;(F,I)]
	[19;(H,G)]		[9;(I,J)]
	[23;(A,B)]		[14;(C,B)]
	[20;(H,I)]		[18;(D,G)]
	[12;(F,J)]		

3. Shortest Path dari A ke F adalah



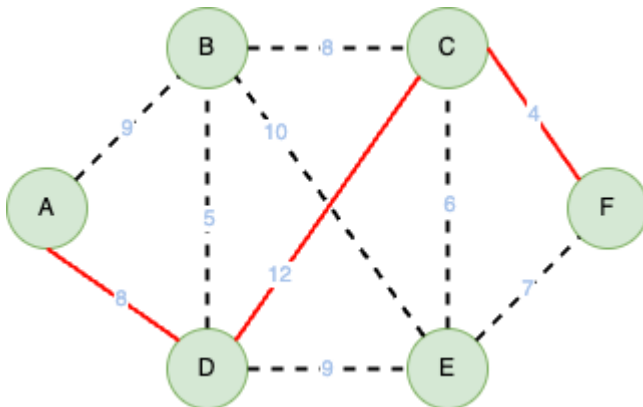
Shortest path metode greedy

$$A-D-B-C-F = 8+5+8+4 = 25$$



Optimal solution

$$A - D - C - F = 8 + 12 + 4 = 24$$



Jadi metode greedy menghasilkan solusi yang cukup baik, tapi bukan yang paling baik bisa dilihat dengan optimal solution. dimana metode greedy membuat keputusan berdasarkan informasi yang **tersedia saat itu** tanpa mempertimbangkan **dampaknya pada solusi secara global atau keseluruhan**.

4. Penggunaan metode Greedy untuk keuntungan maksimal

Pekerjaan	Tenggat Waktu	Keuntungan	D i u r u t k a n b e r d a s a r k a n k e u n t u n g a n :	Pekerjaan	Tenggat Waktu	Keuntungan
A	2	7200		A	2	7200
B	3	6150		I	6	6250
C	4	3550		B	3	6150
D	5	1950		H	4	5600
E	1	2850		C	4	3550
F	3	2700		G	2	3350
G	2	3350		J	4	3200
H	4	5600		E	1	2850
I	6	6250		F	3	2700
J	4	3200		D	5	1950

Jumlah slot yang tersedia : 6 slot (berdasarkan tenggat waktu tertinggi)

Slot	1	2	3	4	5	6
Pekerjaan						
Keuntungan						

Cek acceptance mulai dari pekerjaan dengan keuntungan tertinggi :

I. Pekerjaan A

Tenggat Waktu 2, cek slot 2, kosong, ambil masukkan slot 2

Slot	1	2	3	4	5	6
Pekerjaan		A				
Keuntungan		7200				

II. Pekerjaan I

Tenggat waktu 6, cek slot 6, kosong, ambil masukkan slot 6

Slot	1	2	3	4	5	6
Pekerjaan		A				I
Keuntungan		7200				6250

III. Pekerjaan B

Tenggat waktu 3, cek slot 3, kosong, ambil masukkan slot 3

Slot	1	2	3	4	5	6
Pekerjaan		A	B			I
Keuntungan		7200	6150			6250

IV. Pekerjaan H

Tenggat waktu 4, cek slot 4, kosong, ambil masukkan slot 4

Slot	1	2	3	4	5	6
Pekerjaan		A	B	H		I
Keuntungan		7200	6150	5600		6250

V. Pekerjaan C

Tenggat waktu 4, cek slot 4, terisi, cek slot yang lebih kecil dari 4, slot 1 kosong, ambil masukkan slot 4

Slot	1	2	3	4	5	6
Pekerjaan	C	A	B	H		I
Keuntungan	3550	7200	6150	5600		6250

VI. Pekerjaan G

Tenggat waktu 2, cek slot 2, terisi, cek slot yang lebih kecil dari 2, slot terisi semua, pekerjaan tidak bisa diambil

VII. Pekerjaan J

Tenggat waktu 4, cek slot 4, terisi, cek slot yang lebih kecil dari 4, slot terisi semua, pekerjaan tidak bisa diambil

VIII. Pekerjaan E

Tenggat waktu 1, cek slot 1, terisi, cek slot yang lebih kecil dari 1, slot terisi semua, pekerjaan tidak bisa diambil

IX. Pekerjaan F

Tenggat waktu 3, cek slot 3, terisi, cek slot yang lebih kecil dari 3, slot terisi semua, pekerjaan tidak bisa diambil

X. Pekerjaan D

Tenggat waktu 5, cek slot 5, kosong, ambil masukkan slot 5

Slot	1	2	3	4	5	6
Pekerjaan	C	A	B	H	D	I
Keuntungan	3550	7200	6150	5600	1950	6250

Hitung total keuntungan dari pekerjaan yang diambil :

$$3550 + 7200 + 6150 + 5600 + 1950 + 6250 = 30700$$

REFERENSI

<https://www.youtube.com/watch?v=ZzRi2Qr9lj0>

https://www.youtube.com/watch?v=Ia_Og-qPwhI

https://www.youtube.com/watch?v=LKZ_KglP68Y